IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application. No: 10/676,634	§	Examiner:	Augustine, Nicholas
Filed: October 1, 2003	§		
Inventor(s):	§	Group/Art Unit:	2179
Luis M. Gomes, Madras S. Mohanasundaram	§	Atty. Dkt. No:	5150-82801
	§		
Title: EDITABLE DATA TOOLTIPS	§		
	§		

REQUEST FOR PRE-APPEAL BRIEF REVIEW

Dear Sir or Madam:

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request. This request is being filed with a Notice of Appeal. The review is requested for the reason(s) stated below.

Applicant is in receipt of the Final Office Action mailed April 10, 2009. Claims 1 and 6-23 are pending in the case. Reconsideration of the present case is earnestly requested in light of the following remarks. For brevity, only the primary arguments directed to the independent claims are presented; additional arguments, e.g., directed to the subject matter of the dependent claims, will be presented if and when the case proceeds to Appeal.

Claims 1 and 6-23 were rejected under 35 U.S.C. 103(a) as being unpatentable over Deutscher et al (US Pat. Pub. 2004/0001106, "Deutscher"), in view of Hampapuram et al. (US Pat. Pub. 2004/0221262, "Hampapuram").

Deutscher fails to disclose, **displaying source code for the program on a display during execution of the program, wherein the executing program was compiled from the source code**, as recited in claim 1.

Deutscher is directed to a multimedia presentation production system in which presentation data are separated from presentation logic (see, e.g., [0008]). Per [0009], Deutscher discloses a tool consisting of "graphic user interfaces for easily entering the data associated with the rich media presentation", where "the tool is also designed to make it simple and easy to edit the data of the underlying schema." Per [0017]-[0018], among the various data entry graphical user interfaces (GUIs) disclosed are "data entry grids", which are displayed in a presentation tool window, and "allow the user to enter the information into the presentation data file and the master track media file that is needed to drive the timeline of the presentation. The first of these grids, which is displayed by default once the master track program has

been imported, is the scripts grid. The scripts grid is where the user enters events called script commands that will be triggered during the playback of the presentation."

The Office Action incorrectly equates Deutscher's scripts grid with Applicant's claimed source code of an executing program. Per Deutscher, and as described in more detail in section 2.6.1 ([0137] – [0144]), the scripts grid is "**essentially a scheduling table** where the user enters events (sometimes referred to as script commands) that will be triggered during the playback of the presentation." The entered data in the various data grids (including the scripts grid) are saved to the presentation *data* file, which is then referenced by the presentation system to present the multimedia presentation. A scheduling table is *not* source code that is compilable to generate an executable program, but rather is a means for specifying *input data*..

In direct contrast, as one of skill in the programming arts would readily understand, "source code" is a term of art in the programming domain and refers specifically to program instructions in a programming language that are compiled to produce executable code that may be run on a processor. This is emphasized in the independent claims by the limitation: "**wherein the executing program was compiled from the source code**", which is certainly not the case with the cited script grid. Deutscher does not describe any of the data grids as source code for an executing program. Per cited paragraph 137, the program that is paused to allow for Deutscher's scripts grid editing is "a video or audio program that is designated as the master track", *not* an executable program for which the scripts grid is source code. In the Response to Arguments, the Office Action continues to assert that Deutscher's script grid is equivalent to the source code of claim 1, arguing that "each perform [*sic*] the same functionality". This is not a proper analysis or characterization of the technical content of claim 1. Applicant respectfully reminds the Examiner that *every word in the claim* is to be given consideration, and that the terms of the claim are to be interpreted in light of the Specification. Terms of art should not be re-interpreted contrary to the Specification. A scheduling table (Deutscher's description of the script grid) used by an executing program to manage multi-media presentations does not "perform the same function" as source code that is compiled to generate the executable program. Note that substituting either of these elements for the other in their respective inventions would render the respective inventions inoperable, and so they clearly do *not* perform the same function, and are not equivalent. One of ordinary skill in the programming arts would readily understand that an editing tool and a data file used by an executing program (the presentation viewer) are *not* source code for the executable program, **wherein the executing program was compiled from the source code**, as recited in claim 1. Per Deutscher, the script grid is a data entry grid that allows the user to enter information (i.e., *data*) into the presentation data file, and the presentation data file is (obviously) a *data file*, which is used by the presentation viewer (the executable program) to present a presentation. Neither the script grid, nor the data produced by the script grid, nor

the presentation data file, is compiled into executable code, i.e., executable program instructions, contrary to the Examiner's assertions.

The Examiner further argues that in Deutscher the user can edit the script bar during execution of the program, citing paragraphs [0137] and [0155], as well as Figures 13 and 17. Paragraph [0137], analyzed above, clearly indicates that the scripts grid is a means for providing input data for the presentation viewer (in the form of a presentation data file), *not* source code that can be compiled into executable program instructions. Nor does this citation mention or even hint at the user editing these data *during execution of the presentation viewer*; nor is such data source code for the program (the presentation viewer). Figure 13 shows a text entry dialog box whereby the user can overwrite textual data for the script grid data table, but, as noted previously, the data do not include source code that is compilable to executable program instructions of the presentation viewer. Paragraph [0155] and Figure 17 are directed to editing transcription entries via a text entry dialog box, wherein the user selects a data field, e.g., a time code field, and overwrites the data therein. Again, these citations in no way teach that the transcription entries or the transcription grid in general comprise source code that is compiled to generate the executable program instructions of the presentation viewer, nor that the transcription entries are edited during execution of the presentation viewer. Moreover, in paragraph [0152], Deutscher states that the default language for the transcriptions is *English*, which is decidedly *not* a programming language; the transcription is human-readable text, not program source code. The contents of the grids are data, not source code that is compilable to executable program instructions. As Applicant has explained repeatedly, the transcription text field shown in Figure 17 includes what appears to be a FOR loop designation, but this is actually ***text of a tutorial regarding programming in OLE and Visual Basic***, as may be seen by examining the text of the transcription grid visible behind the data entry dialog box. Thus, the "FOR loop", which Applicant notes includes generic "<code block>" designations, as well, is a textual passage that would be displayed to the user of the presentation viewer while the corresponding audio track of the tutorial is being presented audibly. The cited text is not intended for compilation, nor, Applicant notes, is it compilable at all, as one of skill in the programming arts can readily see. The Examiner continues to argue that this tutorial text is source code for the executing program, which is incorrect. The Examiner appears to assert this (incorrect) equivalence based on certain asserted high-level similarities between the two items. However, this is not the proper way to analyze claim limitations. Note that if one considers any two items from a high enough level of abstraction, *everything* is equivalent, e.g., a porkchop and an apple are both edible objects, but are not equivalent. It is improper for the Examiner to disregard or redefine technical terms in Applicant's claims, thereby refusing to give patentable weight to these terms. Applicant requests that the Examiner give proper weight and meaning to the limitation "displaying source code for the program on a display during execution of the program,

wherein the executing program was compiled from the source code", and submits that if such standard interpretation of these well-known technical terms is used, the Examiner's asserted equivalence will clearly be seen to be incorrect. Hampapuram also fails to teach this claimed feature. Applicant thus submits that the cited art fails to disclose this feature of claim 1.

Nor does the cited art disclose **receiving first user input hovering a mouse cursor over an expression in the source code during execution of the program;** nor **in response to said hovering the mouse cursor over the expression, automatically displaying a GUI element proximate to the expression, wherein the GUI element includes a value of the expression**, as recited in claim 1.

Hampapuram and Deutscher fail to disclose the claimed hover invoked functionality. As discussed above, Deutscher discloses editing a script grid, which, as Deutscher defines it, is "essentially a schedule table", which is not source code as defined in claim 1. Nor is Deutscher's editing of the script grid performed during execution of the program. Applicant further notes that Deutscher teaches displaying the pop-up window in response to user input, specifically, double-clicking on the item to be edited, whereas in claim 1, the GUI element is automatically displayed in response to simply hovering the mouse cursor over the expression. Nor does Hampapuram remedy these deficiencies of Deutscher. For example, per Hampapuram's Abstract, the macro expansions are processed by Hampapuram's tool during the build process of a project ("during a build of a programming project's source files"), where this processing operates to collect and record the macro expansion information into an output file or database. The tool then uses the recorded information to "display the macro expansions in a graphical user interface of the tool, such as for source browsing or viewing static analysis". Applicant notes that "source browsing" and "viewing static analysis" are not run-time operations, and are *nowhere described as being performed while the program is executing*. "Expanding a macro" is not at all the same as automatically displaying a GUI element proximate to the expression (over which the user hovers a mouse cursor *during execution of the program*), where the GUI element includes a value of the expression. More particularly, Hampapuram's GUI does not display the value of an expression during execution of the program, but rather simply displays a macro expansion statically, i.e., *not* at runtime.

Thus, the cited art fails to teach or suggest these features of claim 1.

Nor does the cited art disclose **receiving second user input to the GUI element modifying the displayed value, thereby specifying a new value for the expression; and setting the expression in the program to the new value in response to the second user input, wherein the program continues execution in accordance with the new value of the expression**, as recited in claim 1.

Deutscher nor Hampapuram fail to disclose displaying a value of an expression *in source code of an executing program*, i.e., at runtime, nor modifying the value of such an expression in the source code of an executing program, *where the program continues execution in accordance with the new value of the*

expression. Deutscher's editing of the script grid neither modifies an expression in source code as claimed, nor is performed during execution of the program. Similarly, Hampapuram's macro expansions do not teach or suggest modifying the value of an expression in the source code of an executing program, where the program continues execution in accordance with the new value of the expression, and are not performed during execution of the program.

The Office Action's suggested motivation to combine is incorrect and improper, as explained in detail in the previous Response. There is substantial difference between simply changing a value in a static editor (Deutscher) or expanding a macro in a source code browser or viewer statically (Hampapuram), and dynamically modifying a value of an expression in source code during execution of a program via user input to a tooltip invoked via hovering during runtime. Deutscher edits text in the script grid, which is not a dynamic process performed at runtime; Hampapuram's GUI displays a macro expansion statically, i.e., *not* at runtime. Since neither reference discloses these features, one of ordinary skill in the art would not be compelled to combine them in an attempt to generate Applicant's invention, and so a proper motivation to combine has not been provided, and so the references are not properly combinable. Moreover, even were Hampapuram and Deutscher properly combinable, which they are not, the resulting combination would still not produce Applicant's claimed invention, as explained above.

Thus, for at least the reasons provided above, the cited art of Deutscher and Hampapuram, taken singly or in combination, fails to teach or suggest all the features and limitations of claim 1, and so claim 1, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable. Independent claims 18-21 each includes similar limitations as claim 1, and so the above arguments apply with equal force to these claims. Thus, for at least the reasons discussed above, claims 18-21, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Removal of the section 103 rejection of claims 1 and 6-23 is earnestly requested.

In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-82801/JCH. Also filed concurrently is the following item: Notice of Appeal.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)
Date: 2009-05-04 JCH/MSW